REMARKS/ARGUMENTS

In view of the foregoing amendments and the following remarks, reconsideration of this application is requested. Claims 1-21 are now pending with claims 1, 9, 15, 18, and 21 being independent. Claims 1-3, 6-7, 9-10, 15, 18, and 21 have been amended.

The title of the invention has been amended to make it clearly indicative of the invention to which the claims are directed.

Claims 9 and 18 have been amended in response to the Examiner's objections in the Office Action mailed March 17, 2004.

Claim 21 has been amended in response to the Examiner's rejection under 35 U.S.C. § 112. Applicants have eliminated the limitation of "said second instruction calculates a second return address." Support for claim 21 is shown in Figure 6A and described in the specification on pages 28 and 29.

The Examiner objected to the drawings under 37 CFR 1.83(a). The Examiner states in paragraph 3 of the Office Action mailed March 17, 2004 that the "drawings must show every feature of the invention specified in the claims." The Examiner further states that the drawings and in particular Figure 6B do not show the features of the invention specified in claim 21. Applicants respectfully disagree because support for amended claim 21 is given in Figure 6A and on page 28, lines 13-30. Claim 21 recites determining a first instruction 602 of the first type of instruction executed in a delay slot of a first branch type instruction 601, wherein said first instruction 602 calculates a return address 602b. Claim 21 further recites determining that a second instruction (MV A10,AA4, see page 28) of a second type of instruction is to be executed in a delay slot of the first branch type instruction (B func) as shown in lines 19-24 on page 28. The remaining elements and limitations of claim 21 are shown in lines 25-30 on page 28. Applicants have included new Figures 6C and 6D to illustrate the instruction sequences on page 28. No new matter has been introduced.

The Examiner further objected to Figure 6B in paragraph 4 of the Office Action because "LABEL1 630 should be moved down to correspond with the line pointed to by arrow 634." Applicants have corrected Figure 6B to move LABEL1 down to correspond with the line pointed to by arrow 634.

Claims 1 and 15 have been amended in response to the Examiner's rejections under 35 U.S.C. § 102(a).

As amended, claim 1 describes a digital processing system comprising a microprocessor, wherein the microprocessor performs a method for calling a subroutine that includes a number of steps. The method includes branching to the subroutine by executing a first instruction to provide an address of the subroutine. The method for calling a subroutine also includes calculating a complete return address by executing a second instruction that is next to the first instruction to determine a relative return address, wherein no instructions separate the first and second instructions.

Amended claim 15 describes a method for calling a subroutine in a digital processing system that includes a microprocessor. The method includes the step of branching to the subroutine by executing a first instruction to provide an address of the subroutine. The method also includes calculating a complete return address by executing a second instruction that is next to the first instruction to determine a relative return address, wherein no instructions separate the first and second instructions.

Independent claims 1 and 15 stand rejected under 35 U.S.C. § 102(a) as anticipated by Applicant's Admitted Prior Art (AAPA). Applicants request reconsideration and withdrawal of these rejections for at least the reason that AAPA does not describe or suggest calculating a complete return address by executing a second instruction that is next to the first branch instruction to determine a relative return address, wherein no instructions separate the first and second instructions.

AAPA, in relevant part, as shown in Figure 5 and taught on page 20, lines 17-26 describes a DSP that executes a branch instruction and then forms a return address by executing a move constant instruction (MVK) 502 that moves the least significant half of the address "LABEL" into the least significant half of general purpose register B3. The DSP next executes a move constant instruction high (MVKH) 502 that moves the most significant half of the address "LABEL" into the most significant half of general purpose register B3. AAPA does not describe or suggest calculating a complete return address by executing a second instruction that is next to the first branch instruction, wherein no instructions separate the first and second instructions. In

AAPA, as shown in Figure 5, the MVKH instruction that forms the return address 503 is not next to the branch instruction but follows MVK instruction 502. AAPA also does not describe or suggest calculating a complete return address by executing a second instruction that is next to the first instruction to determine a relative return address. AAPA teaches the use of two instructions to form a complete return address, with the instructions each moving half of the return address to a general purpose register. For at least these reasons, Applicants respectfully submit that claims 1 and 15 are patentable over AAPA.

Claims 4 and 7 depend from independent claim 1. Accordingly, Applicants request reconsideration and withdrawal of the rejections for claims 4 and 7 for the reasons discussed above with respect to claim 1.

Independent claims 1 and 15 stand rejected under 35 U.S.C. § 102(b) as anticipated by Iwao (4,799,151). Applicants request reconsideration and withdrawal of these rejections for at least the reason that Iwao does not describe or suggest calculating a complete return address by executing a second instruction that is next to the first instruction to determine a relative return address, wherein no instructions separate the first and second instructions.

Iwao, in relevant part, as shown in Figure 5 and taught in the Abstract describes a branch and link (BAL) instruction 520(600) that calls a subroutine SUB-S 700. The BAL instruction causes a return address 521(601) from which to restart instruction execution in the main program after completion of the subroutine to be stored on a stack. A return instruction RTN 730 that is executed at the end of subroutine 700 causes the return address 521(601) stored on the stack to be loaded into the program counter and the main program restarts execution at instruction 521(601) specified by the return address. As shown in 750, an offset value α=5 from the return address in the main program may be designated. Thus, the return address in the main program and the offset value can be added to restart instruction execution in a different portion 526(606) of the main program. Iwao does not describe or suggest calculating a complete return address by executing a second instruction that is next to the first instruction to determine a relative return address, wherein no instructions separate the first and second instructions. In Iwao, the return instruction 730 or 750 occurs at the end of the subroutine 700 after a sequence of other instructions have been executed by the processor. Thus, in Iwao, a number of instructions separate the branch instruction BAL from the return instruction and the return instruction that

determines the return address is not next to the first branch instruction BAL. For at least these reasons, Applicants respectfully submit that claims 1 and 15 are patentable over Iwao.

Claims 2-3, 6-7; and 16-17 depend from independent claims 1 and 15, respectively. Accordingly, Applicants request reconsideration and withdrawal of the rejections for claims 2-3, 6-7, and 16-17 for the reasons discussed above with respect to claims 1 and 15.

Dependent claim 5 stands rejected under 35 U.S.C. § 103(a) as obvious over AAPA in view of Hennessy and Patterson, Computer Architecture – A Quantitative Approach, 2nd Edition, 1996 (Hennessy). However, Hennessy does not remedy the failure of AAPA to describe or suggest calculating a complete return address by executing a second instruction that is next to the first branch instruction to determine a relative return address, wherein no instructions separate the first and second instructions. Hennessy shows in Figure 3.1 of page 130 the circuitry for a pipelined datapath with a plurality of stages. On page 134 in Figure 3.4, Hennessy describes pipelining the datapath by adding a set of registers between each pair of pipe stages. The caption to Figure 3.4 describes carrying the PC information from IF/ID register to MEM/WB register through the stages of the pipeline. Figure 3.22 on page 163 shows an alternative pipeline for the datapath of Figure 3.4 in which pipeline stalls are reduced by moving the zero test and branch target calculation into the ID/EX phase of the pipeline from the EX/MEM phase of the pipeline. Hennessy on pages 168-169 describes different schemes to reduce performance penalties caused by branch instructions. In delayed branching, sequential successors are executed whether or not the branch is taken. Hennessy's DLX architecture has one delay slot after each branch instruction shown in Figure 3.27 in which a Branch-delay instruction ($I + 1$) is executed. In Hennessy's architecture, as shown in Figure 3.28 an optimizing compiler can schedule an instruction for the delay slot after examining the dependencies of instructions. Thus, Hennessy does not describe or suggest calculating a complete return address by executing a second instruction that is next to the first branch instruction to determine a relative return address, wherein no instructions separate the first and second instructions.

Calculating a relative return address by executing a second instruction that is next to the first branch instruction with no instructions separating the first and second instructions has several advantages. Use of a single second instruction reduces code size and power dissipation since fewer instructions are fetched. Another advantage as shown in Figure 6A is that the second

instruction ADDKPC provides a relative return address so that the program sequence 600 and subroutine 610 may be relocated to another program address range without changing program sequence 600.

For at least the reasons given above, Applicants respectfully submit that claim 5 is patentable over AAPA in view of Hennessy.

Dependent claim 8 stands rejected under 35 U.S.C. § 103(a) as obvious over AAPA in view of Texas Instruments, TMS32010 User's Guide, 1983 (TI). However, TI does not remedy the failure of AAPA to describe or suggest calculating a complete return address by executing a second instruction that is next to the first branch instruction to determine a relative return address, wherein no instructions separate the first and second instructions. TI shows an Instruction Set Summary in Table 3-2 describing the function of various instructions. For example, the LTA instruction is described as "LTA combines LT and APAC [instructions] into one instruction". TI does not describe or suggest any instructions in Table 3-2 that calculate a relative return address and immediately follow the first branch instruction. Accordingly, Applicants request reconsideration and withdrawal of the rejection of claim 8 for at least the reason discussed above and with respect to claims 1 and 15.

Claims 9 and 18 have been amended in response to the Examiner's rejections under 35 U.S.C. § 103(a).

Amended claim 9 describes a digital processing system comprising a microprocessor, wherein the microprocessor performs a method for forming a relative return address that includes a number of steps. The method includes fetching a sequence of instructions in response to address locations provided by a program counter. The method for forming a relative return address also includes executing a first instruction immediately after executing a branch-to-subroutine instruction in the sequence of instructions by using a first address value provided by the program counter as a source operand, wherein the first instruction forms the relative return address, the relative return address associated with the branch-to-subroutine instruction.

Amended claim 18 describes a method for forming a relative return address in a digital processing system comprising a microprocessor, the method including a number of steps. The method includes fetching a sequence of instructions in response to address locations provided by a program counter. The method for forming a relative return address also includes executing a

first instruction immediately after executing a branch-to-subroutine instruction in the sequence of instructions by using a first address value provided by the program counter as a source operand, wherein the first instruction forms the relative return address, the relative return address associated with the branch-to-subroutine instruction.

Independent claims 9 and 18 stand rejected under 35 U.S.C. § 103(a) as obvious over Hochmuth (5,822,591) in view of Hennessy. Applicants request reconsideration and withdrawal of these rejections for at least the reason that Hochmuth and Hennessy do not describe or suggest the first instruction forming the relative return address, the relative return address associated with the branch-to-subroutine instruction.

Hochmuth, in relevant part, as shown in Figure 1 describes a common software situation in which if variable A is enabled (decision 100), then a set of code associated with variable "A" is executed (box 102). After testing for variable "A" and possibly executing associated code for variable "A", the software tests for a second variable "B" (decision 104) and depending on the result of the second test, the software may execute a set of code associated with variable "B" (box 106). Hochmuth does not describe or suggest the first instruction forming a relative return address, the relative return address associated with the branch-to-subroutine instruction. In Hochmuth, the first instruction 104 does not form a relative return address that is associated with the branch-to-subroutine instruction 100.

Hennessy does not remedy the failure of Hochmuth to describe or suggest the first instruction forming the relative return address, the relative return address associated with the branch-to-subroutine instruction. Hennessey on page 82 describes PC-relative branching in which the destination address of a control flow instruction is specified by supplying a displacement that is added to the program counter. Hennessy does not describe or suggest a first instruction forming the relative return address, the relative return address associated with the branch-to-subroutine instruction. For at least the reasons given above, Applicants respectfully submit that claim 9 and 18 are patentable over Hochmuth in view of Hennessy.

Claims 10, 12; and 19 depend from independent claims 9 and 18, respectively. Accordingly, Applicants request reconsideration and withdrawal of the rejections for claims 10, 12, and 19 for the reasons discussed above with respect to claims 9 and 18.

Dependent claims 11 and 20 stand rejected under 35 U.S.C. § 103(a) as obvious over Hochmuth in view of Hennessy and further in view of TI. However, TI and Hennessy do not remedy the failure of Hochmuth to describe or suggest a first instruction forming the relative return address, the relative return address associated with the branch-to-subroutine instruction. Hennessy on pages 168-170 teaches placement of instructions in delay slots after a branch instruction. Hennessy makes no reference to a first instruction forming a relative return address, the relative return address associated with a branch-to-subroutine instruction.

TI does not remedy the failure of Hochmuth and Hennessy to describe or suggest a first instruction forming a relative return address, the relative return address associated with a branch-to-subroutine instruction. TI, as mentioned above, shows an Instruction Set Summary in Table 3-2 describing the function of various instructions. For example, the LTA instruction is described as "LTA combines LT and APAC [instructions] into one instruction". TI does not describe or suggest any instructions in Table 3-2 that calculate a relative return address, the relative return address associated with a branch-to-subroutine instruction. Accordingly, Applicants request reconsideration and withdrawal of the rejections of claims 11 and 20 for at least the reasons discussed above and with respect to claims 9 and 18.

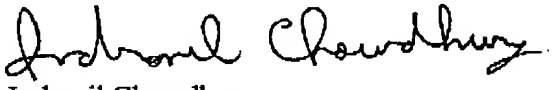
Dependent claim 13 stands rejected under 35 U.S.C. § 103(a) as obvious over Hochmuth in view of Hennessy and further in view of Sharangpani et al. (6,237,077). However, Sharangpani does not remedy the failure of Hochmuth and Hennessy to describe or suggest a first instruction forming a relative return address, the relative return address associated with the branch-to-subroutine instruction. Sharangpani, in relevant part, as described in the Abstract, teaches use of an instruction bundle for processing one or more branch instructions. The instructions are ordered in an execution sequence within the bundle, with the branch instructions ordered last in the sequence. The first branch instruction in the bundle that will be taken is determined and subsequent instructions in the execution sequence are suppressed. Sharangpani does not describe or suggest a first instruction forming a relative return address, the relative return address associated with the branch-to-subroutine instruction. Accordingly, Applicants request reconsideration and withdrawal of the rejection for at least the reasons discussed above and with respect to claim 9.

Dependent claim 14 stands rejected under 35 U.S.C. § 103(a) as obvious over Hochmuth in view of Hennessy and further in view of Haataja (6,137,836). However, Haataja does not remedy the failure of Hochmuth and Hennessy to describe or suggest a first instruction forming a relative return address, the relative return address associated with the branch-to-subroutine instruction. Haataja describes in his Abstract a method of transmitting pictorial data resulting in reduction of required transmission bandwidth by constructing the pictorial data in the form of a composite image of primitive pictures. Figure 8 shows construction of the composite image at a source of pictorial data, and communication of the pictorial data to a portable communicator. The portable communicator may be a paging device, PDA, cellular telephone modified for regenerating the composite image and presenting the composite image on a display. Haataja does not describe or suggest a first instruction forming a relative return address, the relative return address associated with the branch-to-subroutine instruction. Accordingly, Applicants request reconsideration and withdrawal of the rejection for at least the reasons discussed above and with respect to claim 9.

In paragraphs 40-47 of the Office Action mailed March 17, 2004, the Examiner states that Applicant's arguments are not persuasive. Applicants have addressed each of the Examiner's statements in paragraphs 40-47 in addressing the rejections to the claims above.

In view of these remarks and amendments, Applicants submit that this application is now in condition for allowance and the Examiner's prompt action in accordance therewith is respectfully requested. The Commissioner is authorized to charge any additional fees and/or credit any overpayment to Deposit Account 20-0668 of Texas Instruments Incorporated.

Respectfully submitted,



Indranil Chowdhury
Attorney for Applicants
Reg. No. 47,490

Robert D. Marshall, Jr.
Texas Instruments Incorporated
P.O. Box 655474, MS 3999
Dallas, TX 75265
(972) 917-5290

Attachments